| **CSCI 4364/6364 Machine Learning**  **Fall 2023**  Section 81 |  |
| --- | --- |

ML Semester Project Proposal

**Team Name:** AIVerse

**Project Team Members (1-5 members):**

| **Name** | **GWID** |
| --- | --- |
| Toghrul Tahirov | G47609664 |
| Shamil Shukurov | G47855191 |
| Yusif Mukhtarov | G46229570 |
| Aytan Gurbanova | G40711486 |
| Tural Mehtiyev | G33486428 |

**Document Version History**

| **Date** | **Description** |
| --- | --- |
| 10/23/2023 | Initial Draft Submitted **Deadline 10/17/2023** |
|  | Instructor Feedback 1 |
|  | Edits and updates from Feedback 1 |
|  | Approval from instructor |
|  | Final Report Submitted **Deadline 12/14/2023** |

# 1 Analysis

The project is forecasting track from The CityLearn Challenge 2023.

## 1.1 Problem Description:

The purpose is to design regression models to predict the 48-hour-ahead end-use load profiles for three buildings in a synthetic single-family neighborhood as well as the neighborhood-level 48-hour-ahead solar generation and carbon intensity profiles.

The **building-level** target variables include:

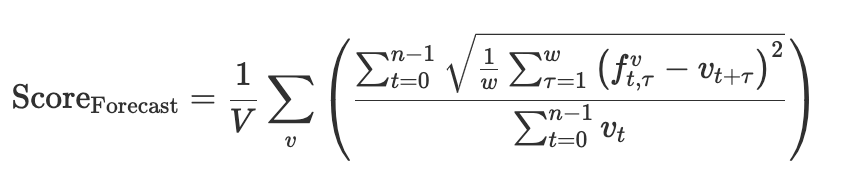
* Cooling Load (kWh)
* DHW Load (kWh)
* Equipment Electric Power (kWh)

The **neighborhood-level** target variables include:

* Carbon Intensity (kgCO2e/kWh)
* Solar Generation (W/kW)

## 1.2 Performance Criteria:

The forecast track score, ScoreForecast, is the average over all of the variables being forecast, of the normalized mean root mean square error (RMSE) of the forecasts made.



Where:

* t: Environment time step index;
* n: Total number of time steps, t, in 1 episode;
* τ: Forecasting window time step index;
* w: Length of forecasting window (48hrs);
* b: Total number of buildings;
* v: Forecasting variable;
* V: Total number of variables to forecast (3b + 2),
* ft,τv: Forecast of variable v for time step t+τ, made at time t;

## 1.3 Related Work:

Time series analysis has been used for many decades. Various machine learning methods have been tried and proven useful, ranging from classical statistical models to deep learning models, to forecast future figures.

Energy consumption forecasting is a popular ML problem that has many papers dedicated to it. In [[1]](#_heading=h.umw6umhtj0z9), they have built SVM and ARIMA models for energy consumption forecasting, and the SVM models outperformed ARIMA. In another study [[2]](#_heading=h.kqvpu4cnoq3j), the authors proposed a novel learning procedure, named *Anchor-based Forecasting* motivated by the anchored-based object detection methods. In the described method, the objective is to predict bounding boxes of specific objects. According to the authors, in this method the network is given predefined boxes (anchors) and the problem is transformed into predicting the offset instead of the actual load value. This method simplifies the task by making it easier for the network to learn.

Along with classical ML models, many Deep Learning models have also been implemented and performed well in predicting energy load. For example, for short-term load forecasting, a CNN and RNN based method was presented in [[3]](#_heading=h.8gahp5xyhcr). Additionally, a RNN based sequence to sequence (Seq2Seq) model was presented in another study [[6]](#_heading=h.cubodl8b1imi), both of which resulted in good performance.

For better forecasting, many hybrid approaches were also implemented. One of them combines attributes from ensemble forecasting, artificial neural networks and deep learning architectures [[4]](#_heading=h.kwej9f14bh4q). In this study, to forecast a week-ahead load, the model initially clusters input data using a novel fuzzy clustering method. A regression model is applied to each cluster. The model follows a two-stage process, first, a radial basis function neural network (RBFNN) is trained using three-fold cross-validation and the hidden layers of the best three RBFNNs are used to transform the input data to a four dimensional dataset. Then, a convolutional neural network (CNN) is deployed receiving as input the latter dataset. Another work [[5]](#_heading=h.9u36a5kaplu7) implemented a methodology with LSTM, CNN and auto-encoder to train only one model for forecasting many time series. Their model has outperformed Temporal Convolutional Network model. The authors highlighted that although statistical methods such as SARIMAX and Prophet showed good performance, they require training multiple models, one for each time series in the dataset.

## 1.2 Project Objective:

The objective of the project is to develop regression models capable of accurately predicting the 48-hour-ahead end-use load profiles for individual buildings and neighborhood-level solar generation and carbon intensity profiles in a synthetic single-family neighborhood. The project aims to optimize these predictions, ultimately achieving a high-performance score, to enhance energy efficiency and sustainability in the neighborhood.

# 2 Hypothesis

## 2.1 Methodology

Forecasting the future, especially 48 hours ahead, is inherently challenging. While predicting the future with certainty is impossible, modeling and estimating it based on past patterns is feasible. We seek to leverage the seasonality and stationary inherent to the data to make accurate forecasts. Below, we provide a list of both the Statistical Machine Learning and Deep Learning models chosen for this task, alongside the rationale for their inclusion.

### 2.1.1 Statistical Machine Learning Models

* ARIMA and SARIMA: Autoregressive Integrated Moving Average and Seasonal Autoregressive Integrated Moving Average are conventional methods for time series forecasting. They perform optimally on stationary data and can capture trend and seasonality. These models will essentially serve as a foundation and provide a benchmark for evaluation of advanced models.
* Statistical/Neural Prophet: This duo is an evolution from traditional ARIMA models and is an example of the early anchor-based approaches. While the statistical variant employs a generalized additive model (GAM) to fit data, its neural counterpart utilizes a neural network with the same core philosophy.
* Gradient Boosting Algorithms: Predominantly tree-based models, they are adept at handling large datasets and a variety of feature types. The fact that these models incorporate many weak-learners ensures a reduction in overfitting, while their boosting mechanism focuses on errors, which improves the forecast accuracy.

### 2.1.2 Deep Learning Models

* LSTM (Long Short-Term Memory): A type of recurrent neural network (RNN) explicitly designed for time series and sequences.
* TCN (Temporal Convolutional Network): Adapts CNNs for sequential data and offers advantages over RNNs in terms of parallelism and longer memory.
* Transformers (Temporal Fusion Transformer - TFT): The TFT leverages the self-attention mechanism from transformers, making it capable of focusing on different time steps of the input sequence, capturing long-term dependencies and relationships in the data.

## 2. 2. Data Description

* Source: NeurIPS 2023 Citylearn Challenge
  + Number of examples: 720 hourly samples for each of the 3 buildings and 2 neighborhoods level datasets
  + Features: The data consists of features pertinent to building level energy loads, carbon intensity, and solar generation.
  + Labels: Continuous labels corresponding to the variables indicated in Problem Description

Here, both variables will be forecasted collectively for the entire neighborhood.

Note: We confirm that the data used for this project is neither confidential, private, nor proprietary and will be shared as part of the final delivery.

## 2. 3 Experiment

To evaluate the performance of our models, we'll employ a rolling forecast origin methodology, ensuring our model's adaptability to new data points. Additionally, we will split the data into training and test sets to validate the model's performance on unseen data.

## 2.4 Metrics

* Mean Absolute Error (MAE): Represents the average error magnitude.
* Root Mean Squared Error (RMSE): Emphasizes larger errors.
* Mean Absolute Percentage Error (MAPE): Provides a relative error measure.

Through rigorous testing and comparison between the models, we aim to pinpoint the most suitable approach for our 48-hour forecast.

# 3 Synthesis

**Software & Tools**

Python: The primary programming language for the project, offering a wide range of libraries and tools tailored for data science and machine learning.

Pandas & Numpy: For data loading, manipulation, and numerical operations.

Scikit-learn: For preprocessing tasks and model evaluation metrics.

TensorFlow & Keras: To construct and train regression prediction models. Hyperparameter tuning will be facilitated using Keras Tuner.

Matplotlib & Seaborn: For data visualization, plotting patterns, and model evaluation metrics.

Google Colab: As our primary development environment for interactive coding, model training with GPU support, and sharing.

GitHub: For version control, collaboration, and maintaining a clear history of model iterations and changes.

**Hyperparameter Tuning Approach:**

* We'll utilize TensorFlow's Keras Tuner to define and search the hyperparameter space. This tool provides a range of methods, including RandomSearch, Hyperband, and BayesianOptimization.
* We'll select the most suitable method based on our dataset.
* Once the tuning process identifies the best hyperparameters, we'll validate the performance on a separate validation set.
* The final step involves training the model on the full dataset using the selected optimal hyperparameters.

**Project Work Plan with Tasks and Expected Dates**

1. Oct 21 - Oct 22: Data Understanding & Pre-processing.
2. Oct 23: Feature Engineering.
3. Oct 24 - Oct 26: Model Development.
4. Oct 27 - Oct 28: Hyperparameter Tuning.
5. Oct 29 - Oct 30: Model Evaluation & Iteration.
6. Oct 31: Documentation & Reporting.

# 4 Validation

## 4.1 Results:

* When computing the raw results for each experiment in the time series forecasting analysis, we will follow the established evaluation criteria as per the project description. The score for the forecast track, denoted as ScoreForecast, will be computed as the average over all the variables being forecast. This score will be determined based on the normalized mean root mean square error (RMSE) of the forecasts made.
* For the purpose of visualizing and tabulating the results in the final report, we intend to include the following components. Line plots will display the actual and predicted values for each forecasting variable over time, allowing for a clear comparison of the model's performance. Moreover, Tensorboard will be utilized to showcase the learning as loss function curves, providing a visual representation of how the loss function converges over multiple training iterations, demonstrating the model's learning progress.
* Raw results table will contain a detailed breakdown of the computed raw results for each experiment, including the actual values, predicted values, and corresponding RMSE scores for each variable and building, facilitating a comprehensive assessment of the model's performance.
* Summary statistics table will present a summary of key statistical metrics, such as mean, standard deviation, and median, for both the actual and predicted values across the forecasting window, aiding in the quick comparison of the model's accuracy and precision.

The more detailed information about tables and visualization methods are described below:

* Time Series Plots: Time series plots will effectively display the actual and predicted values over time for each variable. This allows for a comprehensive understanding of the model's performance in capturing trends, patterns, and seasonal fluctuations.
* Box-and-Whisker Plots: These plots will help visualize the distribution of forecast errors for different variables. By comparing the medians, quartiles, and outliers of the forecast errors, one can gain insights into the consistency and accuracy of the forecasting model.
* Forecast Error Distribution Plots: Visualizing the distribution of forecast errors for each variable will help identify any patterns or anomalies in the model's predictive accuracy. Histograms and kernel density plots will be used to illustrate the spread and skewness of the forecast errors.
* Error Metrics Table: This table will include various error metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) for each variable and building. Such a table provides a detailed comparison of the model's performance across different metrics, enabling a comprehensive evaluation of forecasting accuracy.
* Summary Statistics Table: This table will summarize the statistical properties of the forecasted values and actual values, including mean, standard deviation, median, minimum, and maximum values. This helps in understanding the central tendencies and variations in the data, providing insights into the overall behavior of the variables over the forecasting window.

## 4.2 Conclusions:

In the future, we will determine the success of our research hypothesis through a statistical test comparing the distribution of predicted values with the actual values. We will utilize the Mean Absolute Percentage Error (MAPE) as the primary metric for comparison. Following this, we will conduct a paired t-test to quantify the significance of the differences between the actual and predicted values. The resulting p-value will provide insights into the effectiveness of our chosen forecasting methodologies. If the p-value is less than 0.05 (just an assumption, we will modify this value as necessary), we will conclude that there are significant disparities between the forecasts and actual values, indicating a need for model refinement. Conversely, if the p-value is greater than 0.05, we will conclude that our forecasting models accurately predict the values, reinforcing their reliability for practical applications. These formal conclusions will serve as essential evidence of the suitability and accuracy of the forecasting models.

# 5 References

###### D. Liu, Q. Chen and K. Mori, "Time series forecasting method of building energy consumption using support vector regression," *2015 IEEE International Conference on Information and Automation*, Lijiang, China, 2015, pp. 1628-1632, [doi: 10.1109/ICInfA.2015.7279546](https://ieeexplore.ieee.org/document/7279546).

###### M. Tzelepi, P. Nousi and A. Tefas, "Improving Electric Load Demand Forecasting with Anchor-Based Forecasting Method," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: [10.1109/ICASSP49357.2023.10096754](https://ieeexplore.ieee.org/document/10096754/).

###### Hosein Eskandari, Maryam Imani, Mohsen Parsa Moghaddam, Convolutional and recurrent neural network based model for short-term load forecasting, Electric Power Systems Research, Volume 195, 2021, 107173, ISSN 0378-7796, <https://doi.org/10.1016/j.epsr.2021.107173>.

###### Sideratos, George & Ikonomopoulos, Andreas & Hatziargyriou, Nikos. (2020). A novel fuzzy-based ensemble model for load forecasting using hybrid deep neural networks. Electric Power Systems Research. 178. 106025. 10.1016/j.epsr.2019.106025.

###### Rodney Rick, Lilian Berton, Energy forecasting model based on CNN-LSTM-AE for many time series with unequal lengths, Engineering Applications of Artificial Intelligence, Volume 113, 2022, 104998, ISSN 0952-1976, <https://doi.org/10.1016/j.engappai.2022.104998>.

###### Zhao, Dongbo & Ge, Qian & Tian, Yuting & Cui, Jia & Xie, Boqi & Hong, Tianqi. (2020). Short‐term load demand forecasting through rich features based on recurrent neural networks. IET Generation, Transmission & Distribution. 15. 10.1049/gtd2.12069.